# 11.JUnitTestforReversinga Word

Aim:

TowriteaJUnittestthatverifiesthefunctionalityofamethodthatreverses agivenword. The JUnit test will use assert statements to check if the word is reversed correctly.

Algorithm:

1. **Step 1:**Createamethodtoreversetheinputword.
   * UseaStringBuildertoreversethestring,asitprovidesanefficientwayto reverse the word.
2. **Step2:**WriteaJUnittest classtovalidate thereversefunctionality.
   * Thetest classwillusethe@Testannotation tomarktest methods.
   * UsetheassertEqualsmethodfromJUnittocheckiftheoutputmatchesthe expected reversed string.
3. **Step3:**Definesampletestcasesforvariousinputwords,includingnormal,edge cases like an empty string, and a single character.
4. **Step4:**Executethetestsandvalidatethe output.

CodeImplementation:

**Step 1: reverse.java**

**package** ex2.strrev;

**publicclass** reverse {

**public** String reverseWord(String word) {

**if** (word == **null**) {

**returnnull**; // Return null for null input

}

StringBuilder reversed = **new** StringBuilder(word);

**return**reversed.reverse().toString(); // Return the reversed string

}

}

**Step 2 : reverseTest.java**

**package** ex2.strrev;

**importstatic** org.junit.jupiter.api.Assertions.\*;

**import** org.junit.jupiter.api.Test;

**class** reverseTest {

@Test

**publicvoid** testReverseNormalWord() {

reverse r = **new** reverse();

*assertEquals*("olleh", r.reverseWord("hello"));

}

@Test

**publicvoid** testReverseEmptyString() {

reverse r = **new** reverse();

*assertEquals*("", r.reverseWord(""));

}

@Test

**publicvoid** testReverseSingleString() {

reverse r = **new** reverse();

*assertEquals*("a", r.reverseWord("a"));

}

@Test

**publicvoid** testReverseNull() {

reverse r = **new** reverse();

*assertNull*(r.reverseWord(**null**));

}

}

Sample Input:

# TestCase1:

* + - Input:"hello"
    - ExpectedOutput: "olleh"

# TestCase2:

* + - Input:""(EmptyString)
    - ExpectedOutput:""(EmptyString)

# TestCase3:

* + - Input:"a"
    - ExpectedOutput: "a"

# TestCase4:

* + - Input:null
    - ExpectedOutput: null

# SampleOutput:

* + **TestCase1:**
    - Input:"hello"
    - Output:"olleh"

# TestCase2:

* + - Input:""(EmptyString)
    - Output: "" (EmptyString)

# TestCase3:

* + - Input:"a"
    - Output:"a"

# TestCase4:

* + - Input:null
    - Output:null

Results:

Theoutput foreach test caseshouldmatch theexpectedresults:

* + **TestCase1:**Thestring"hello"wasreversedcorrectlyto"olleh",andtheassertion passed.
  + **TestCase2:**Theemptystring""returnedtheexpectedoutputof"",andtheassertion passed.
  + **TestCase3:**Thesinglecharacter"a"remainedunchanged,asexpected,andthe assertion passed.
  + **TestCase4:**Thenull input correctlyreturnednull,andtheassertion passed.

In conclusion, the JUnit tests confirm that the reverseWord method works correctly for differentinputtypes,includingnormalwords,emptystrings,singlecharacters,andnull values.



